

Pembuatan Aplikasi Delivery Order dengan Menggunakan HTML5 Berbasis Mobile

Greffika Exstrilia¹, Justinus Andjarwirawan², Lily Puspa Dewi³

Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) - 8417658

E-mail: clarettagreff@gmail.com¹, justin@petra.ac.id², lily@petra.ac.id³

ABSTRAK

Di tengah banyaknya aktifitas yang terus meningkat, dan banyak menyita waktu masyarakat untuk membeli makan langsung ke tempat makan atau depot yang diinginkan, serta tidak sempat untuk mencuci pakaian dan lain-lain. Maka beberapa depot makanan serta tempat *laundry* melalui contoh kasus yang ada, melahirkan suatu solusi dengan menyediakan layanan pesan dan kirim makanan serta mengambil kemudian mengantarkan ke tempat pemesan. Kebanyakan depot makanan serta tempat *laundry*, yang telah memakai layanan pesan antar pada tempat makan dan ambil antar pada tempat *laundry*, menggunakan sistem panggilan via telepon atau mengirimkan pesan singkat. Apabila terdapat contoh kasus, jika seseorang tersebut tidak tahu berapa nomor telepon depot makanan yang menyediakan layanan pesan antar, serta tidak tahu menu apa saja yang tersedia, maka pastilah seseorang tersebut terpaksa keluar rumah dan mencari depot makanan itu serta memastikannya secara langsung.

Dengan menggunakan aplikasi, dengan internet yang terinstal pada perangkat *gadget* pada sistem operasi Android, layanan pesan antar ini dapat diakses lebih praktis dan penggunaannya sangat mudah, serta keuntungan lainnya dapat menghemat waktu dan biaya. Aplikasi sistem pesan antar makanan dan *laundry* ini, merupakan sistem informasi dengan berbasis *mobile device* yang dirancang dengan HTML5 dan dibantu kompilasi oleh *phoneGap*, yang menyediakan proses pemesanan menu makanan pada tempat makan serta *laundry* pada cakupan area tertentu, yang didukung oleh *Google Maps* yang bertujuan untuk mempermudah melacak posisi tempat makan dan *laundry* serta mengoptimalkan layanan pesan antar dan ambil antar untuk makanan dan *laundry*. Proses pengiriman makanan dan *laundry* dilakukan secara manual oleh kurir yang bertugas.

Dengan demikian pelanggan yang menggunakan sistem ini, dapat lebih mudah melakukan proses pemesanan serta dapat menghemat tenaga serta biaya.

Kata Kunci: Pesan Antar, *Mobile device*, HTML5, *PhoneGap*, Google Map

ABSTRACT

In the midst of increasing activities and public's time-consuming, people prefer to buy food directly to the food stalls which are desired and when the people do not have time to wash clothes and others, they prefer to take the clothes to the laundry place. But, the problem is people are too busy and having limited time to go to buy food and taking clothes to the laundry place. From the existing problem above, food stalls and laundry places invent a solution by providing delivery order service. The delivery order service is effective and efficient. It can send ordered food and take-dropping laundry to the customer. Most food stalls and

laundry places that use already the delivery order service use the systems that are via phone calls and sending short messages. If there is a case, such as, when customers don't know the phone number is and don't know what the menu is available then surely they will have to go out of the house and looking for a place to eat and confirm the menu directly.

By using the internet applications in the gadget which installed on the Android operating system, Short Message Service can be accessed more practically and very easy to use as well as other benefits that can save time and costs. Delivery order application is using an information system with mobile-based device that is designed with HTML5 and assisted by compilation by PhoneGap, which provides the process of ordering food which can show the food stalls menu and laundry places on the scope of a particular area, which will be powered by Google Maps which aims to make it easier to track the position of food stalls and laundry places and to optimize the delivery order for food and laundry. The process of delivering food and laundry is done manually by couriers who are in duties.

Thus, the customers who use this system will more easily do the delivering process and can save energy and costs.

Keywords: *Delivery Order, Mobile device, HTML5, PhoneGap, Google Maps.*

1. PENDAHULUAN

Pada era telekomunikasi, perkembangan teknologi komunikasi mengarah ke sebuah teknologi yang berbasis *mobile*. Saat ini *mobile phone* tidak hanya digunakan sebagai media berkomunikasi untuk telepon dan sms, tetapi juga sebagai media untuk berkomunikasi langsung melalui internet untuk mengirim dan menerima data. Hal ini menyebabkan tingginya minat terhadap sebuah *mobile phone* yang sangat pintar, karena sangat menunjang bagi penggunaanya dalam mengatasi kesibukan dan kebutuhan yang dimilikinya.

Masyarakat pada masa sekarang ini dalam kehidupan sehari-hari lebih memilih untuk mendapatkan suatu informasi dan layanan dengan cara yang instan dan cara yang tidak perlu membuang waktu, dengan menggunakan berbagai aplikasi *mobile* yang sudah tersedia pada *smart phone*, maka media lainnya seakan-akan tersisihkan.

Dengan menggunakan teknologi tersebut, penulis merancang sebuah aplikasi yang cukup unik dan efektif dalam mengerjakan pelayanan transaksi pesan antar atau distribusi *Delivery Order* berbasis *mobile* dimana aplikasi pemesanan ini sebagai *client* dan dapat berjalan pada *platform* Android yang dapat dimanfaatkan bagi masyarakat sekitar Siwalankerto dan untuk seluruh civitas akademika Universitas Kristen Petra, mahasiswa atau staf dan

dosen. Aplikasi *Delivery Order* ini dibagi menjadi 2 kategori layanan yaitu *Food Delivery Order* (Pesan Antar Makanan) dan *Laundry Delivery Order* (Pesan Antar Laundry).

Kedua layanan ini menolong pengguna untuk menjangkau tempat makan dan tempat *laundry* dengan mudah, praktis serta mengoptimalkan waktu dan tenaga. Kedua layanan ini dikumpulkan dari berbagai macam depot makanan dan tempat *laundry* yang berada pada daerah Siwalankerto atau area sekitar Universitas Kristen Petra, yang terukur dari sepanjang Jalan Siwalankerto, sebagai batasan *tracking* wilayah untuk kedua layanan *Delivery Order* ini. Serta dengan adanya aplikasi ini dapat membantu membangun perekonomian masyarakat yang secara khusus memiliki mata pencaharian diantara kedua layanan yang tersedia dalam aplikasi ini, dan bersedia bekerjasama untuk saling mendukung aplikasi ini agar dapat diterima dan dioperasikan dengan baik oleh pengguna.

2. LANDASAN TEORI

2.1. HTML5 (*Hypertext Markup Language*)

HTML5 [5] adalah sebuah bahasa markah untuk menstrukturkan dan menampilkan isi dari *World Wide Web*, sebuah teknologi inti dari Internet.

Berikut adalah kategori fitur utama yang berada pada *HTML5* :

- (1) *Core HTML5*. Bagian ini dari *HTML5* membuat spesifikasi resmi dari versi W3C. Ini mencakup *semantic elements* yang baru dari *HTML5*, *Web form widget* yang baru dan ditingkatkan, dukungan audio dan *video*, dan *canvas for drawing* dengan *JavaScript*.
- (2) *Features that were once HTML5*. Fitur-fitur ini muncul dari *HTML5* spesifikasi asli yang telah dipersiapkan oleh WHATWG. Sebagian besar adalah spesifikasi untuk fitur yang membutuhkan *JavaScript* dan untuk mendukung aplikasi *Website*. Yang paling signifikan adalah lokal data *storage*, *offline application*, dan *messaging*.
- (3) *Feature that are sometimes called HTML5*. Ini adalah fitur generasi berikutnya yang sering disamakan dengan *HTML5*, meskipun tidak pernah menjadi bagian dari standar *HTML5*. Kategori ini berisi *CSS3* dan *geolocation* [6].

2.2. Google Maps

Google Maps [2] merupakan sebuah layanan peta dunia *virtual* berbasis *website* yang disediakan oleh *Google*. *Google Maps* menawarkan peta yang dapat digeser (*panned*), diperbesar (*zoom in*), diperkecil (*zoom out*), dan dapat diganti dalam beberapa mode (*map*, *satelit*, *hybrid*, dan lain-lain) Selain itu, terdapat juga fitur untuk menambahkan *Google Maps* dalam web dengan menggunakan *Google Maps API*. *Google Maps API* adalah *library JavaScript*. Dengan menggunakan *Google Maps API*, dapat menghemat waktu dan biaya untuk membuat aplikasi peta digital yang handal.

Beberapa sintaks yang sering digunakan pada *Google Maps* yaitu:

- (1) *Google.maps.LatLng*
Merupakan sintaks yang digunakan untuk menunjuk pada lokasi di peta. *LatLng* memiliki banyak kegunaan pada *Google Maps API*. Contoh sintaks *google.maps.LatLng*:

```
var myLat lng = new
Google.maps.LatLng (myLatitude,
myLongitude)
```

- (2) *Google.maps.Map*

Merupakan *classJavaScript* yang merepresentasikan sebuah peta. Contoh sintaks *google.maps.Map*:

```
var map = new
google.maps.Map
(document.getElementById
("map_canvas"), myOptions);
```

- (3) *Google.maps.Marker*

Membuat sebuah *marker* atau penanda pada pilihan tertentu. Bila sebuah peta spesifik, penanda diletakkan pada peta pada saat *construction*. Contoh sintaks *google.maps.Marker*:

```
var marker = new
google.maps.Marker (opts?)
```

- (4) *getZoom()* dan *setZoom()*

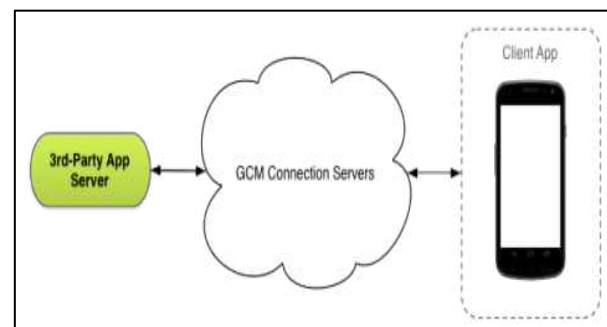
getZoom() mengembalikan sebuah nilai yang mengidentifikasi nilai dari *zoom level* yang sekarang *setZoom (zoomLevelnumber)* mengatur tingkat *zoom* peta. Contoh *getZoom()* dan *setZoom()*:

```
var zoomLevel = map.getZoom () ;
Map.setZoom (12) ;
```

2.3. Teknologi Push Notifikasi (Google Cloud Messaging)

Google Cloud Messaging [4] untuk Android adalah layanan yang dapat membantu *developer* untuk mengirim data dari server ke aplikasi Android buatan mereka pada *device* Android setiap kali data baru tersedia. Mengintegrasikan GCM pada aplikasi Android kita dapat meningkatkan pengalaman pengguna dan menyimpan banyak tenaga baterai *device* Android.

Aplikasi Android yang menggunakan GCM tidak perlu berjalan di belakang layar untuk menerima sebuah notifikasi. GCM (dalam hal ini Layanan *Google Play*) akan membangun aplikasi tersebut dengan mekanisme *Broadcast Intent* (*Event* yang terjadi saat ada notifikasi baru) ketika ada pesan tiba selama aplikasi tersebut diatur untuk diizinkan menerima Pesan Siaran. Namun tidak sedikit pula aplikasi berjalan di latar belakang yang menggunakan GCM *Service*. Berikut akan ditampilkan Gambar 1 sebagai model arsitektur *Google Cloud Messaging*



Gambar 1. Arsitektur Google Cloud Messaging

Ini adalah bagaimana komponen-komponen GCM saling berinteraksi [9] :

- (1) *Google GCM Connection Server* mengambil pesan dari server aplikasi dan mengirimkannya ke aplikasi *client*.
- (2) Pada server aplikasi, menerapkan HTTP dan protokol XMPP untuk berkomunikasi dengan GCM *Connection Server*. Server aplikasi mengirim pesan ke *server* koneksi GCM; yang *enqueues* koneksi *server* dan menyimpan pesan, kemudian mengirimkannya ke aplikasi *client*.
- (3) *Client App* adalah aplikasi *client* GCM-enabled. Untuk menerima pesan GCM, aplikasi ini harus mendaftar dengan GCM dan mendapatkan identifikasi unik yang disebut *registration token*. Jika menggunakan koneksi *server* XMPP, aplikasi *client* dapat mengirim pesan “*upstream*” kembali ke *server* aplikasi

Berikut contoh segmen program *send notification* menggunakan GCM

```
function send_message($gcmid,$message)
{
    $gcm = new GCM();
    $registatoin_ids = array($gcmid);

    $result=$gcm>send_notification($registatoin_ids,
    $message);
    echo $result;
}
```

2.4. MySQL

MySQL [8] merupakan sebuah *database open-source*. MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi MySQL juga memiliki lisensi yang dapat dibeli untuk penggunaan komersial. MySQL dibuat sebagai *Database Management System* (DBMS) untuk *relational database*. Sebagai *relational database*, MySQL menggunakan tabel-tabel untuk menyimpan informasi. Meskipun membutuhkan perencanaan yang lebih mendalam untuk desain dan pembuatan *databasenya*, MySQL menjamin realibilitas dan integritas data.

Pada MySQL untuk dapat membuat *database*, tabel *field-field* ataupun mengubah dan menghapus, digunakan *Structured Query Language* (SQL).

Berikut adalah beberapa fungsi dasar pada SQL [7] :

1. Membuat *database*, digunakan fungsi: *CREATE DATABASE* nama_database;
2. Memilih *database*, digunakan fungsi: *USE* nama_database;
3. Membuat tabel pada *database* yang sudah dipilih: *CREATE TABLE* nama_tabel (nama_kolom1 tipe_data, nama_kolom2 tipe_data);
4. Mengkonfirmasi tabel yang sudah dibuat: *SHOW TABLES*;

5. Melihat isi kolom dari sebuah tabel: *SHOW COLUMN* from nama_tabel; atau *DESCRIBE* nama_tabel;
6. Memasukkan data pada tabel yang sudah dibuat: *INSERT INTO* nama_tabel (kolom_1, kolom_2,...) *VALUES* (value_1, value_2,...)
7. Melihat data yang sudah dimasukkan pada tabel: *SELECT* nama_kolom1, nama_kolom2,... from nama_tabel; Atau *SELECT *from* nama_tabel *WHERE* kondisi;
8. Mengubah data yang ada pada tabel: *UPDATE* nama_tabel *SET* nama_kolom=nilai *WHERE* kondisi;
9. Menghapus data yang ada pada tabel: *DELETE* from nama_tabel *WHERE* kondisi;

2.5. PhoneGap

PhoneGap [1] adalah *open source framework* untuk membuat *cross-platfrom native applications* menggunakan teknologi *web* mulai dari HTML, CSS, dan JavaScript. Tipe dari aplikasi ini disebut sebagai *hybrid application*. *PhoneGap* diciptakan untuk mempermudah *mobile development*. *PhoneGap* bekerja dengan cara merubah *web application package* menjadi *native application*. Aplikasi yang telah dibuat akan ditampilkan dalam bentuk *web view* yang memungkinkan pengguna untuk melakukan interaksi dengan aplikasi tersebut.

Kompatibilitas *PhoneGap* dengan *Mobile OS* yang ada saat ini dapat dilihat pada Tabel 1.

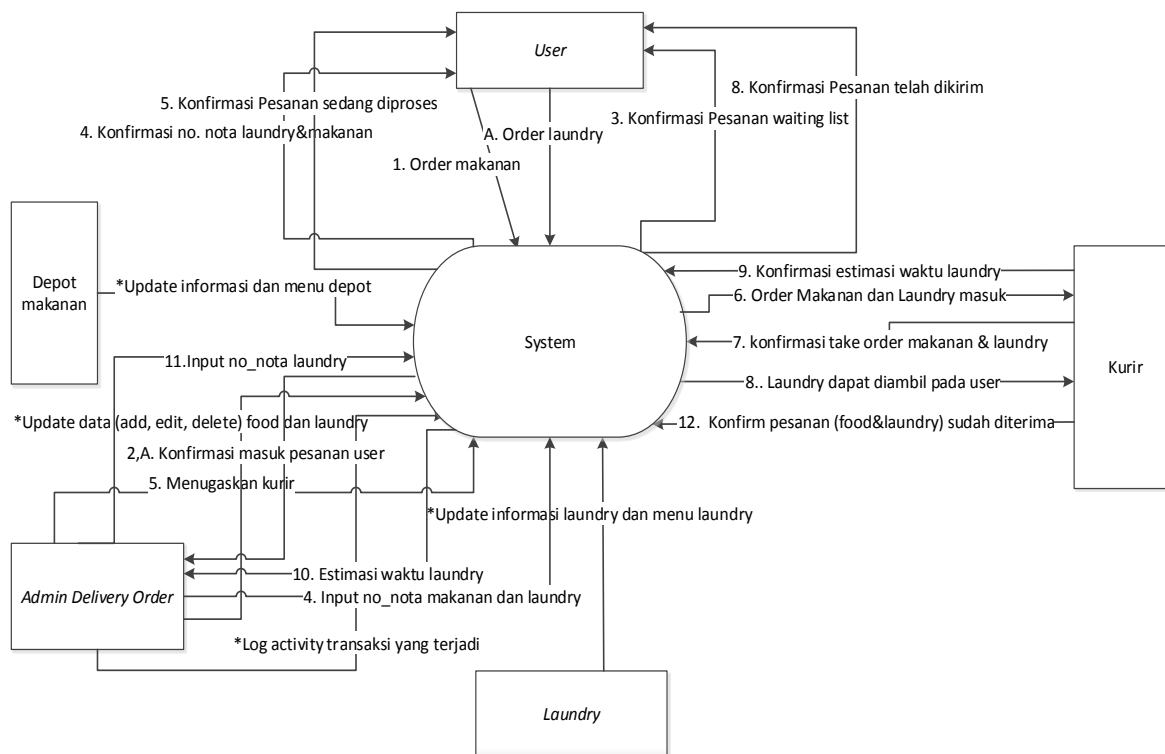
Tabel 1 Kompatibilitas PhoneGap

Features	iOS	An-droid	Win-dows Phone	Black-berry	Symbian	WebOS
Accelerometer	Yes	Yes	Yes	Yes	Yes	Yes
Camera	Yes	Yes	Yes	Yes	Yes	Yes
Compass	Yes	Yes	Yes	No	No	No
Contacts	Yes	Yes	Yes	Yes	Yes	No
File	Yes	Yes	Yes	Yes	No	No
Geolocation	Yes	Yes	Yes	Yes	Yes	Yes
Media	Yes	Yes	Yes	No	Yes	No
Network	Yes	Yes	Yes	Yes	Yes	Yes
Notification	Yes	Yes	Yes	Yes	Yes	Yes

2.6. jQuery Mobile

jQuery Mobile [3] adalah *web framework* yang dioptimalkan untuk perangkat-perangkat layar sentuh (*touch-optimized*). sebagai *library*, *jQuery Mobile* yang dikenal sebagai *library* JavaScript maupun *framework mobile* ini dikembangkan oleh tim *jQuery Project*. Tujuan pengembangannya adalah berfokus pada pembuatan *framework* yang kompatibel dengan beraneka ragam perangkat *smartphone* dan tablet,

jQuery Mobile merupakan sistem *user interface* berbasis HTML5 dan pengembangan dari *library* JQuery dan jQuery UI yang sedang populer saat ini. kelebihan jQuery adalah kodenya ringan, menerapkan desain yang dapat beradaptasi dengan berbagai jenis perangkat *mobile*, dan menonjolkan *markup* semantik. Oleh karena itu, desainnya sangat fleksibel dan mempunyai tema yang bisa disesuaikan.



Gambar 2. Context Diagram Delivery Order

Berikut adalah contoh Segmen Program dari penggunaan jQuery Mobile

```

<a onclick="tekanBack()" class="ui-btn-
left ui-btn ui-btn-inline ui-mini ui-
corner-all ui-btn-icon-left ui-icon-

```

3. DESAIN SISTEM

Sistem *Delivery Order* ini terbagi menjadi 5 pengguna, diantaranya adalah admin *delivery order*, user, depot makanan, laundry, dan kurir. Bila digambarkan dalam bentuk *context diagram* berikut gambaran singkatnya pada Gambar 2.

Cara kerja sistem pada *context diagram* diatas dapat diuraikan dalam bentuk penjelasan singkat, yaitu:

(1) User

Melakukan *take order* yang berupa makanan atau laundry pada system. Setelah itu beberapa saat kemudian muncul pemberitahuan dari sistem jika pesanan menunggu konfirmasi (*waiting list*), setelah admin merespon pesan user, maka system akan mengenerate notifikasi yang memberitahukan kepada user jika pesanan sedang dalam proses. Hingga user mendapatkan nomor nota dari sistem untuk dimasukkan pada halaman menu *bill's number* yang bertujuan untuk mengetahui detail pesanan tersebut. Selanjutnya jika pesanan sudah siap diantar, maka sistem akan mengirimkan pemberitahuan kepada user berupa kalimat, pesanan sudah dikirim.

(2) Admin Depot Makanan

Hanya memiliki hak akses pada aplikasi untuk *edit* depot makanannya sendiri, serta menggunakan fungsi *add*, *edit*, *delete* menu makanan.

(3) Admin Laundry

Memiliki hak akses yang sama dengan laundry pada aplikasi, yaitu untuk edit *laundrynya* sendiri, serta menggunakan fungsi *add*, *edit*, *delete* menu *laundry*.

(4) Kurir

Hak akses yang diberikan kepada kurir untuk menggunakan aplikasi adalah kurir dapat melihat dan memilih (*pick order*) order makanan dan laundry yang perlu untuk dikonfirmasi oleh kurir, kemudian konfirmasi kembali ketika kurir sudah selesai menyelesaikan pengirimannya.

(5) Admin Delivery Order

Admin *delivery order* ialah admin yang bertanggung jawab atas proses jalannya aplikasi ini. Admin *delivery order* bertugas untuk input data depot makanan, laundry serta detail menu dari masing-masing layanan. Kemudian memberi konfirmasi pesanan user.

4. PENGUJIAN SISTEM

Pada bab ini akan dibahas tentang pengujian sistem sesuai dengan analisis dan desain *system*. Pengujian sistem meliputi pengaturan *database*, MySQL dan implementasi aplikasi. Dari hasil pengujian yang dilakukan, menghasilkan analisis dan dibuat kesimpulan untuk pengembangan aplikasi selanjutnya. Spesifikasi *device* Android yang digunakan pada pengujian system dibagi menjadi beberapa pengujian diantaranya ialah *Device* pengujian yang digunakan, *Operating System* dari *device* yang digunakan, *Display Size device* tersebut, CPU yang normalnya mampu mengoperasikan *device* yang diuji, kemudian yang terakhir *Memory Internal* pada *device* yang dipakai untuk pengujian. Pengujian ini memakai 3 device yaitu 1 Laptop dan 2 *Handphone* dengan spesifikasi yang berbeda. Selengkapnya dapat dilihat pada Tabel 2.

Tabel 2. Tabel spesifikasi device pengujian

Device	Operating System	Display Size	CPU	Internal Memory
Sony VAIO E Series 14P	Windows 7	720 x 1280 pixels (~294 ppi pixel density)	Qualcomm MSM8916 Snapdragon 410, Quad-core 1.2 GHz Cortex-A53	8 GB storage, 2 GB RAM
Samsung Galaxy S6 Edge	Android 4.4.2. (Kitkat)	800 x 1280 pixels (~189 ppi pixel density)	Exynos 4412 Quad, Quad-core 1.6 GHz Cortex-A9	16 GB storage, 2 GB RAM
Sony Xperia C3 D2533	Android OS, v4 4.2 (KitKat)	720 x 1280 pixels (~267 ppi pixel density)	Quad-core 1.2 GHz Cortex-A7	8 GB, 1 GB RAM

Pada bagian ini akan membahas alur *user* terhadap aplikasi. *User* dapat melihat beberapa bagian menu utama pada halaman *index*, yaitu halaman *Main Menu*, *Food*, *Laundry*, *Tracking Map Food and Laundry*, *Bill's Number*, *Recent Order*, dan *Favorites*.



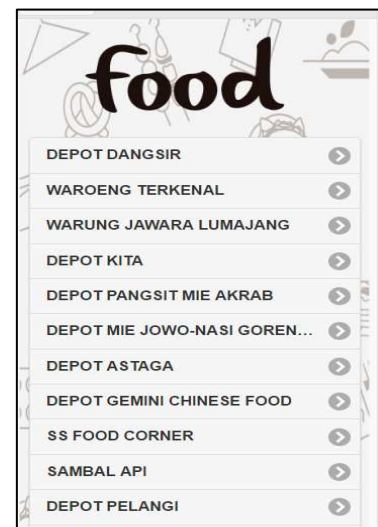
Gambar 3. Halaman Index Main Menu

Pada Gambar 3 halaman bagian *main menu*, adalah menu utama dari aplikasi *delivery order* ini. Pada halaman *main menu* ini terdapat 6 menu utama di antaranya ialah *laundry*, *food*, *tracking*, *bill's number*, *recent order* dan *favourite*. Aplikasi *delivery order* ini memiliki nama yaitu *Deli-O* yang memiliki arti *Delivery Order* beserta *icon* yang sangat menggambarkan bahwa aplikasi *delivery order* ini mengerjakan layanan pesan antar makanan dan *laundry*.



Gambar 4. Halaman Index Bagian Menu Laundry

Pada Gambar 4 bagian ini, akan ditampilkan beberapa *laundry* yang berhasil dikumpulkan oleh admin disekitar jalan Siwalankerto. Namun, *laundry* ini jumlahnya tidak menentu, karena sewaktu-waktu admin dapat menambahkan atau bahkan bisa mengurangi *laundry* yang terdapat pada *list laundry* pada halaman Menu *Laundry*. Hal ini disebabkan situasi dan kondisi dari *laundry* tersebut Untuk memilih *laundry user* hanya membutuhkan sekali sentuh pada *laundry* manapun yang *user* inginkan, berikutnya akan diarahkan ke halaman detail dari *laundry* tersebut.



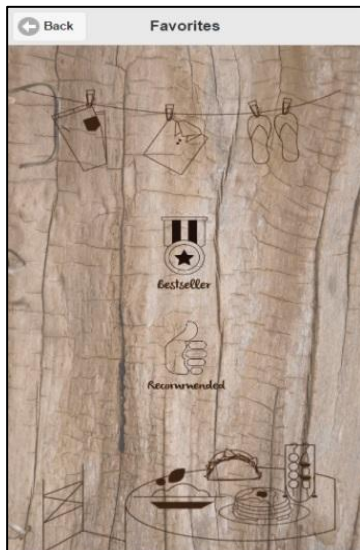
Gambar 5. Halaman Index Bagian Menu Food

Pada Gambar 5 dapat dilihat terdapat beberapa macam depot makanan yang berhasil dirangkum oleh admin di wilayah Jalan Siwalankerto. Namun, depot makanan ini jumlahnya tidak menentu, karena sewaktu-waktu admin dapat menambahkan atau bahkan bisa mengurangi depot makanan yang terdapat pada *list depot* pada halaman Menu *Food*. Hal ini disebabkan situasi dan kondisi dari depot makanan tersebut. Untuk memilih depot makanan *user* hanya membutuhkan sekali sentuh pada depot manapun yang *user* inginkan, berikutnya akan diarahkan ke halaman detail dari depot makanan tersebut.



Gambar 6. Halaman Index Menu Tracking Map Food dan Laundry

Halaman *index* bagian menu *tracking map food* dan *laundry* pada Gambar 6 merupakan halaman *tracking map* yang berfungsi sebagai, pelacak cepat letak dan posisi beberapa depot makanan dan *laundry* yang berada di sekitar Jalan Siwalankerto, tampilan *map* dibatasi dengan cakupan area khusus Jalan Siwalankerto, Surabaya. Pada *map* terdapat *list mark* gambar *icon* biru dan merah muda yang adalah penanda posisi depot makanan atau *laundry* tersebut berada. *Icon* biru menandakan *icon* untuk posisi *laundry* dan *icon* merah muda menandakan *icon* untuk posisi depot makanan berada.



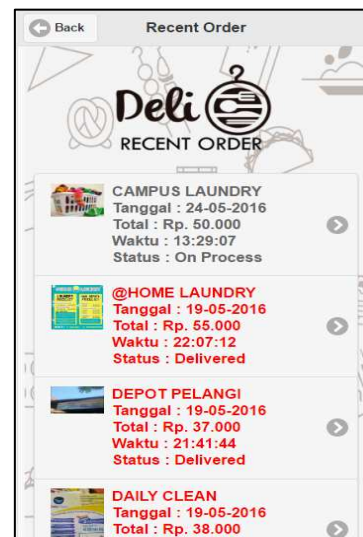
Gambar 7. Halaman Index Bagian Menu Favourites

Halaman *index* bagian menu *favourites* pada Gambar 7 ini terdiri dari 2 menu utama yaitu *best seller* dan *recommended*. Keduanya dimiliki juga oleh depot makanan dan *laundry*. *Best Seller* diartikan sebagai penjualan terbaik dan disini akan terlihat penjualan paling favorit yang diminati *user* dengan ditandai *rating* bintang, kemudian untuk *recommended* diartikan sebagai penjualan yang direkomendasikan oleh admin *delivery order*.



Gambar 8. Halaman Index Bagian Menu Bill's Number

Pada Gambar 8 merupakan tampilan dari *Bill's Number* yang mana *user* dapat memasukkan nomor nota pada kolom yang disediakan lalu klik *icon search* untuk diarahkan pada halaman detail dari nomor nota yang dimasukkan.



Gambar 9. Halaman Index Bagian Menu Recent Order

Pada Gambar 9 Halaman *index* bagian menu *recent order*, memuat beberapa pesanan *user* (depot makanan dan *laundry*) dari mulai yang terbaru hingga pesanan yang sudah lama. Untuk pesanan yang *high light* merah merupakan pesanan yang telah dikirim oleh kurir, dan yang berwarna hitam masih menunggu (*waiting*) atau sedang dalam proses (*on process*).

5. KESIMPULAN DAN SARAN

Dari proses perancangan, pembuatan, dan hasil pengujian aplikasi Deli-O, *Delivery Order* dapat diambil kesimpulan sebagai berikut:

5.1. Kesimpulan

- Tampilan aplikasi pada *mobile*, menurut hasil pengisian kuesioner oleh responden menyatakan 93% baik. Namun, untuk lebih menarik perhatian responden diperlukan untuk mengupdate tampilan sesuai dengan

trend yang sedang digunakan dalam dunia teknologi dan informasi yang tentunya *user friendly*.

- Kemudian, kejelasan informasi dan fitur yang ada, menurut hasil pengisian kuesioner oleh responden menyatakan 91% baik. Dalam hal ini, sangat diperlukan untuk menjelaskan informasi secara rinci agar konten dalam fitur yang ada dapat diterima dengan baik.
- Poin ketiga ialah kelengkapan informasi, menurut hasil pengisian kuesioner oleh responden menunjukkan 90% adalah baik. Langkah konkrit yang dilakukan ialah lebih memperhatikan informasi dengan cara melengkapi konten yang telah ada, supaya responden nantinya memahami informasi yang tersedia dengan baik.
- Berikutnya adalah, kemudahan menggunakan aplikasi atau *user friendly*. Menurut hasil pengisian kuesioner oleh responden menunjukkan 100% sangat baik.
- Manfaat aplikasi, menurut hasil pengisian kuesioner oleh responden menyatakan 100% sangat baik.
- Poin terakhir ialah penilaian aplikasi secara keseluruhan. Menurut hasil pengisian kuesioner oleh responden menunjukkan 98% baik. Artinya adalah responden telah memahami cara kerja dari aplikasi ini walaupun belum sepenuhnya dan aplikasi ini dapat digunakan dengan baik sebagaimana mestinya.

5.2. Saran

- Aplikasi ini dapat dikembangkan lagi untuk proses *delivery order* diluar makanan ataupun *laundry*.
- Cakupan areanya juga bisa diperluas supaya jangkauan *customernya* lebih besar dan lebih banyak mendapatkan data sebagai objek *delivery order* yang baru.
- Menambah fitur lain yang lebih menarik dan lebih lengkap, misalnya fitur *cleaning service*, *print and photocopy*, dan lain sebagainya. Mungkin bisa menjadi nilai tambah aplikasi ini jadi lebih baik.
- Kelengkapan informasi yang dimuat dalam aplikasi, dapat didetailkan kembali agar informasi yang ingin disampaikan dapat diterima dengan baik oleh *user*.
- Cara kerja sistem yang ada, mungkin dapat diperbarui dengan menyesuaikan kemajuan teknologi dan

informasi atau menggunakan inovasi cara kerja sistem yang lebih *user friendly* sehingga dapat memudahkan bagi *customer* mendatang ketika mengoperasikan aplikasi Deli-O ini.

6. DAFTAR PUSTAKA

- [1] Andrew, L. 2011. *PhoneGap: Beginner's Guide*. Birmingham: Packt Publishing.
- [2] Chopvitayakun, S. 2015. *Android Application to Enhance Performance of Internship Program Implementing Cloud Computing Platform and Infrastructure*. Retrieved November 25, 2015, Hal 2530-2538 from <http://www.sciencedirect.com/science/article/pii/S1877042815046364>.
- [3] Firtman, M. 2012. *jQuery Mobile: Up and Running*. California: O'Reilly Media.
- [4] Google Cloud Developer. 2016. Google Cloud Messaging: Overview. Retrieved February 5, 2016, from <http://developers.google.com/cloud-messaging/gcm#arch>
- [5] Kadir, A. 2013. *From Zero to a Pro HTML5 Panduan untuk Mempelajari Pengembangan Rich Internet Applications melibatkan CSS, Javascript, dan PHP*. Yogyakarta: Andi Offset.
- [6] MacDonald, M. 2014. *HTML5: The Missing Manual, 2nd Edition*. Sebastopol: O'Reilly Media, Inc.
- [7] McLaughlin, B. 2011. *PHP and MySQL: the Missing Manual*. California: O'Reilly Media.
- [8] MySQL. 2016. *Top Reasons to Use MySQL*. Retrieved March 10, 2016, from: <http://www.mysql.com/why-mysql/topreasons.html>
- [9] Narayanan, C. (n.d.). Retrieved May 30, 2015, from http://ultraimager.webs.com/Seriousandroiddeveloper/GCM_tutorial.pptx